Experement1: Familiarization with DEBUG

Done by: Abeer(www.UOB-BH.com)

Equipments:

- **An IBM compatible PC**
- **Debug program (available on all PC Windows)**

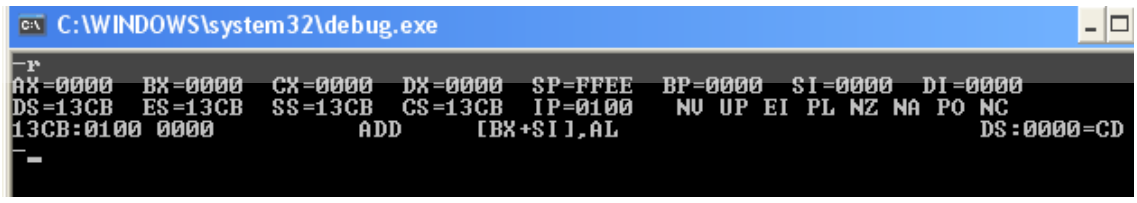
Objectives:

The objectives of this experiment are to get familiar with Debug and to:

- **Examine and modify the contents of the 8086's internal registers.**
- **Examine and modify the contents of the memory (code, data and stack).**

Procedure:

1. **Turn on the PC, click on the Start button, choose Run, then type: DEBUG ↵**
2. **DEBUG responds with the hyphen, - , prompt. The hyphen prompt indicates that DEBUG is ready to accept commands.**
3. **Type? ↵ At the DEBUG prompts (-), this will print a summary of all valid commands.**
4. **You can quit from DEBUG any time by entering Q**
5. **Restart DEBUG program**
6. **Type R ↵ (REGISTER command) to display the contents of all 8086 internal registers. To what values the registers are set ? what values did the contents of all 8086 internal registers.**



7. **Calculate the physical address (PA) of the next instruction to be executed.**

[CS: IP] >> 13CB:0100

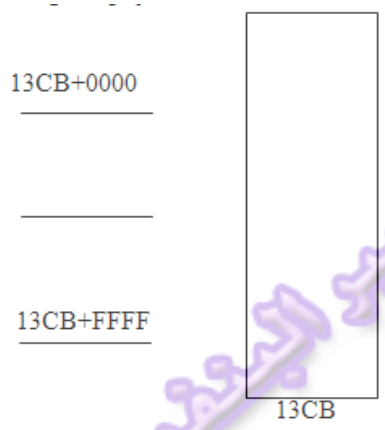
13CB0+00100=13DB0

8. **Calculate the physical address of the current top of the stack.**

   [SS: SP] >> 13CB: FFEE

   13CB0+FFEE= 23C9E

9. **Draw a memory map which shows the active memory segments, with the lowest and highest physical address of each segment.**



10. **Use R or r command followed by any of the general purpose registers or segment registers to change the content of that register (you can even change IP and CS contents).**

    We choose to change the contain of BX , it changes from 0000 to 1234

```
-R
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=13CB  ES=13CB  SS=13CB  CS=13CB  IP=0100   NV UP EI PL NZ NA PO NC
13CB:0100 0000          ADD     [BX+SI],AL                      DS:0000=CD
-R BX
BX 0000
:1234
-R
AX=0000  BX=1234  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=13CB  ES=13CB  SS=13CB  CS=13CB  IP=0100   NV UP EI PL NZ NA PO NC
13CB:0100 0000          ADD     [BX+SI],AL                      DS:1234=00
-_
```

11. **Display the current contents of BP then modify its value to (9AB)$_{16}$.**

```
-R BP
BP 0000
:9AB
-R BP
BP 09AB
```

12. **Display the contants of the flag registor and then change the state of the status flags to their complement values.**

```
-RF
OV UP DI NG ZR AC PO CY   -_
```

13. **Use the Dump command (D) to display the first 128 bytes of the current data segment. Use D again to display the next 128 bytes.**

```
-d
13D2:0100  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:0110  00 00 00 00 00 00 00 00-00 00 00 00 34 00 C1 13   ............4...
13D2:0120  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:0130  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:0140  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:0150  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:0160  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:0170  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-d
13D2:0180  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:0190  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:01A0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:01B0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:01C0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:01D0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:01E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:01F0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-
```

14. **With the ENTER (E) command, load the first double word of the current location in data segment, with the value 456789AB in little endian form. Before terminating the command , verify that the memory contents have been change by stepping back through the memory locations by pressing the (hyphen) key.**

```
-R
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=13D2  ES=13D2  SS=13D2  CS=13D2  IP=0100   NV UP EI PL NZ NA PO NC
13D2:0100 0000           ADD     [BX+SI],AL                   DS:0000=CD
-D
13D2:0200  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:0210  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:0220  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:0230  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:0240  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:0250  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:0260  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
13D2:0270  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-D DS:13D2
13D2:13D0          AB 89 67 45 00 00-00 00 00 00 00 00 00 00     ..gE..........
```

15. **WITH THE enter (E) COMMAND, LOAD AT OFFSET EA=300 of the current data segment the following bytes.**

```
0B20:0300  E2.49   F2.54   2E.43   C7.45   06.20   B2.33   90.34   04.31
0B20:0308  00.20   5D.3d   5F.20   5B.49   2E.6e   8B.74   0E.72   B0.6f
0B20:0310  90.64   2E.75   A1.63   B2.74   90.69   2E.6f   8B.6e   36.20
0B20:0318  B4.74   90.6f   2E.20   8B.4d   16.69   B6.63   90.72   2E.6f
0B20:0320  8A.70   1E.72   B8.6f   90.63   F8.65   C3.73   50.73   26.6f
0B20:0328  8B.72
-D
0B20:0100  74 38 3C 0D 74 34 3A 06-1E 96 74 2E 3A C3 74 2A   t8<.t4:...t.:.t*
0B20:0110  3C 3A 74 03 E9 5F FF 80-3E 0C 98 02 34 00 0F 0B   <:t.._..>...4...
0B20:0120  00 EB D9 46 EB 14 E9 4D-FF BA ED 89 E9 CB E5 BA   ...F...M........
0B20:0130  17 8B E9 C5 E5 4E 5F 9D-F9 C3 4E EB 51 80 CF 01   .....N_...N.Q...
0B20:0140  81 CD 00 80 E8 12 E2 46-E8 E4 DF 74 0D E8 45 00   .......F...t..E.
0B20:0150  AC E8 41 00 81 CD 00 40-EB 34 3C 0D 75 09 B0 00   ..A....@.4<.u...
0B20:0160  AA 81 CD 00 40 EB CE E8-2B 00 E8 EC DF 06 57 51   ....@...+.....WQ
0B20:0170  0E 07 BF 0B 8F B9 06 00-81 CD 00 40 F2 AE 75 0B   ...........@..u.
-D Ds:300
0B20:0300  49 54 43 45 20 33 34 31-20 3D 20 49 6E 74 72 6F   ITCE 341 = Intro
0B20:0310  64 75 63 74 69 6F 6E 20-74 6F 20 4D 69 63 72 6F   duction to Micro
0B20:0320  70 72 6F 63 65 73 73 6F-72 07 A9 02 00 75 05 2E   processor....u..
0B20:0330  FF 06 B0 90 2E 80 3C 00-75 19 A9 01 00 75 09 2E   ......<.u....u..
0B20:0340  C7 06 B2 90 02 00 EB 0E-50 B0 03 B4 FF E8 9E 00   ........P.......
0B20:0350  58 EB 03 E8 1D 01 58 C3-F9 C3 55 51 26 8A 4F 08   X.....X...UQ&.O.
0B20:0360  32 ED 0B C9 74 0D 8D 6F-09 E8 19 04 73 08 E8 0E   2...t..o....s...
0B20:0370  00 E2 F6 F9 EB 06 2E 89-2E C5 90 F8 59 5D C3 26   ............Y].&
```

16. Using D command, find out the characters whose ASCII codes have been entered in the previous step.

```
ITCE 341 = Intro
duction to Micro
processor....u..
......<.u....u..
........P.......
X.....X...UQ&.O.
2...t..o....s...
............Y].&
```

17. Dump the last 16 bytes of the ROM (FFFF: 0000) to find the BIOS date of the PC.

```
-D FFFF:0000 L 16
FFFF:0000  EA D1 FF 00 F0 30 36 2F-30 39 2F 31 30 20 FC A7   .....06/09/10 ..
FFFF:0010  34 12 00 00 00 00                                 4.....
-
```

18. Execute a SEARCH (S) command to determine which locations in the range F000:0000 to F000: FFFF have the ASCII codes of “BIOS”.

```
-S F000:0000 FFFF "A"
```

```
F000:E623
F000:E62F
F000:E640
F000:E643
F000:E74E
F000:E751
F000:E992
F000:E99F
F000:E9C8
F000:E9E7
F000:EA10
F000:EA14
F000:EA46
F000:EA76
F000:EC88
F000:ECE1
F000:EE1C
F000:EECD
F000:F2C5
F000:F6FC
F000:F758
F000:FE85
F000:FF07
F000:FFDF
```